

クリッカブルムービーとそのオーサリングツール

藤本 英明[†] 吉岡 信夫[†] 大村 皓一^{††}

[†]大阪工業大学

^{††}宝塚造形芸術大学

本研究は、プレイヤーにおいてインタラクティブに動画をクリックできるクリッカブルムービーを提案する。既にある動画に、XML ファイルを結びつけることでクリッカブルムービーにする。その XML のフォーマットとして、CMML(Clickable Movie Markup Language) を提案する。プレイヤーは CMML ファイルを読んで、クリッカブル領域のある動画を表示する。動画のクリッカブル領域がクリックされると、リンクされたテキストや画像が表示される。オーサリングツールは、複雑なタグ文章を書かずに、インタラクティブに CMML ファイルを作成できるようにする。

Clickable Movie and its authoring tool

Hideaki Fujimoto[†] Nobuo Yoshioka[†] Kouichi Ohmura^{††}

[†]Osaka Institute of Technology

^{††}Takarazuka University of Art and Design

In this study, we propose the Clickable Movie. On its player, we can click movie interactively. Other than an existing movie file, a XML file is associated. As its XML format, we propose the CMML(Clickable Movie Markup Language). The player read in the CMML file and displays the movie file with clickable region. When the region on the movie is clicked, a linked text or image is displayed. The authoring tool enables to make CMML file interactively, instead of writing the complex tag text.

REST AVAILABLE COPY

1. はじめに

動画をデジタルで保存することが簡単になり、保存された動画が増えてくると思われる。

従来は、保存された動画は、そのまま再生されるだけだった。つまり、動画は、情報が孤立していて、見ることしかできなかった。そこで、動画と情報を結びつけて、動画をインタラクティブに扱えるようにする。その方法として、動画をクリックできるようにする。クリックすると新しい情報を得られるようにすることで、動画と情報を結びつける。その結果、動画を見るだけでなく、一つの情報源として扱えるようになる。

本研究では、既にある動画を有効に利用するために、動画を加工しないで新しい情報を得られるようにする。動画を直接加工しないで新しい情報を得られるようにするため、XML形式のファイルを別に作ることにする。

XML形式のファイルには、どの領域を何時クリックできるようにするかを決める領域と時間、クリックされたときに表示する情報などを保存することにする。

これらの情報を保存したXMLファイルを参照し、クリッカブルムービーとして再生できるプレイヤーを作成する。また、動画を再生しながらXML形式のファイルを作成できるオーサリングツールを作成し、XML形式の付加情報を簡単に作成できるようにする。

2. Clickable Movie Markup Language

クリッカブルムービーを実現するためのXMLとして、Clickable Movie Markup Language(CMML)を提案する。CMMLの定義は付録に収録した。

CMMLは、クリッカブル領域を指定した時間に表示する為の情報や、その領域がクリックされたときに表示する為の情報を記述することを目的に定義した。クリッカブル領域は、複数の領域を一つの領域として扱ったり、時間とともに移動することができるようにした。複数の領域を一つの領域として扱うことで、複雑な形のクリッカブル領域を実現し、無駄なタグを省くことができる。また、クリッカブル領域を時間とともに移動させることで、動きのある画像に対して効果的なクリッカブル領域を設定することができる。

これらの条件を満たすために、CMMLのタグは次のような構成にした。

`<cmml></cmml>` ... CMMLの開始と終了を表すタグ。

`<moviefile href="パス"><title>タイトル</title></moviefile>` ... cmmlタグの中に記述し、パスは動画のあるパスを、タイトルは動画のタイトルを表す。

`<clickable></clickable>` ... cmmlタグの中に記述し、クリッカブル領域と表示する時間、クリックされたときに表示する情報を記述する。

`<start></start>` ... clickableタグの中に記述し、クリッカブル領域と表示の開始時間を表す。

`<region></region>` ... startとendタグの中に記述し、クリッカブル領域の領域の座標を表す。このタグを、複数書くことで、複雑な形のクリッカブル領域を実現することが出来る。また、時間的に変化しないクリッカブル領域の場合は、endタグの中の記述を省略しても良い。

`<left></left>,<top></top>,<right></right>,<bottom></bottom>` ... regionタグの中に記

述し、領域の左上の x 座標、y 座標、領域の右下の x 座標、y 座標の値を記述する。また、全ての値を 0 にすることで、そのクリッカブル領域が移動しないことを表すことができる。

`<time></time>` ... start と end タグの中に記述し、クリッカブルタグの表示時間を表す。start タグの中では、クリッカブルの開始時間、end タグの中ではクリッカブルの終了時間を記述する。

`<hour></hour>`, `<min></min>`, `<sec></sec>`, `<msec></msec>` ... time タグの中に記述し、それぞれ、動画の開始位置からの時間、分、秒、ミリ秒の値を記述する。

`<end></end>` ... clickable タグの中に記述し、クリッカブル領域と時間の終わりの値を示す。end タグに含まれる要素は、start タグと同じである。また、end タグの中で、region タグが設定されてない場合は、領域は固定され、設定されている場合は、start の region から end の region まで領域が移動する。

`<info href="パス">INFO</info>` ... clickable タグの中に記述し、クリックされたときに表示する情報を記述する。パスは画像のパス、INFO は文字の情報である。情報は図 1 のような情報ダイアログとして表示される。



図 1 情報ダイアログ

次に、簡単な CMML の例を示す。複数の領域を一つのクリッカブルタグに記述した CMML である。

```
<cmml>
  <moviefile href="Globe.avi"><title>CMML サンプル 1 </title></moviefile>
  <clickable>
    <start>
      <region>
        <left>208</left><top>100</top><right>248</right><bottom>178</bottom>
```

```

</region>
<region>
<left>248</left><top>95</top><right>267</right><bottom>172</bottom>
</region>
<time>
<hour>0</hour><min>0</min><sec>1</sec><msec>972</msec>
</time>
</start>
<end>
<region>
<left>63</left><top>109</top><right>101</right><bottom>175</bottom>
</region>
<region>
<left>101</left><top>124</top><right>175</right><bottom>227</bottom>
</region>
<time>
<hour>0</hour><min>0</min><sec>3</sec><msec>452</msec>
</time>
</end>
<info href="bird.bmp">複雑な領域</info>
</clickable>
</cmml>

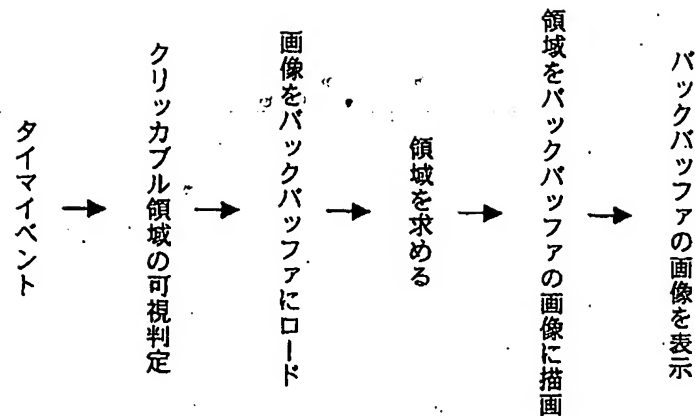
```

この例では、(208,100) (248,178)と(248,95) (267,172)の矩形を合成した領域が、(63,109) (101,175)と(101,124) (175,227)の矩形を合成した領域まで移動することを表している。このように、一つのクリッカブルタグの中に複数の領域を書くことで、同じ付加情報を持った複雑な形のクリッカブル領域を実現し、タグの記述を最小限に押さえることが出来る。

3. プレイヤ

CMML プレイヤは、CMML に従って動画をクリッカブルムービーとして再生する。クリッカブルムービーは、動画を CMML で指定した時間と領域だけクリックできるようにする。また、その領域がクリックされたときに情報をダイアログで表示する。クリッカブルムービーは、従来の動画の再生に、クリッカブル領域の表示、クリックされたときの当たり判定、付加情報の表示を加えたものである。

CMML プレイヤは、タイマイイベントを用いて動画の再生時間の管理をしている。また、このタイマイイベントでクリッカブル領域の表示も行っている。CMML プレイヤはタイマイイベントにおいて、次のような手順でクリッカブル領域の表示と動画の再生を行う。



タイマイイベントが発生すると、画像とクリックカブル領域を表示する。ただし、クリックカブル領域の表示は、ちらつきがないようにバックバッファにある動画にクリックカブル領域を描画してから画面に表示する。また、移動するクリックカブル領域は、start と end の region と time を使い、直線的に移動するとして、再生時間での領域を求めている。

クリックの当たり判定は、クリックされた点が、有効なクリックカブル領域に含まれるかを調べる。判定の対象を有効なクリックカブル領域のみにすることで、クリックカブルタグの数の影響を受けずに判定できる。また、複数の領域があるクリックカブルタグの場合は、どれかがクリックされたと判定されたときに、それらの領域への判定を終わることで、無駄な判定を行わないようにしている。判定の結果、クリックされた点が有効なクリックカブル領域に含まれていた場合は、付加情報をダイアログで表示する。

付加情報は、info タグの情報と、href で指定された画像をダイアログで表示する。また、ダイアログを開くときに、ウィンドウが開くようなアニメーションを加えることで、動画から情報が飛び出すような演出をしている。

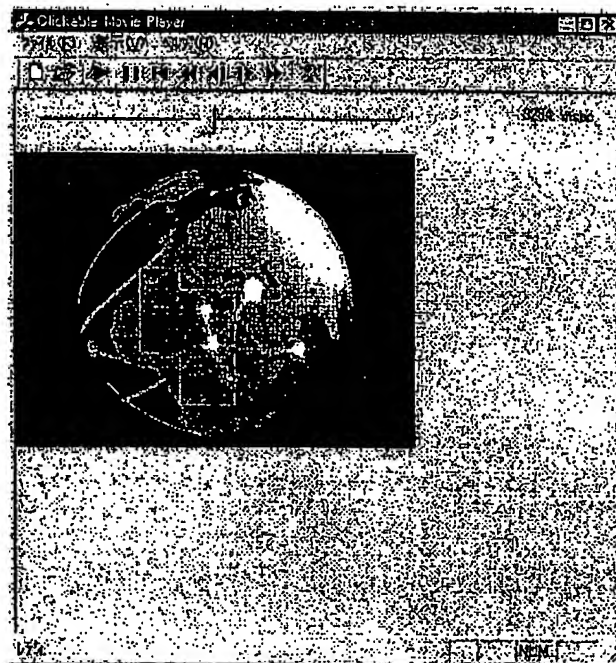


図2. CMML プレイヤの実行画面

4. オーサリングツール

CMML ファイルはテキストデータであるので、テキストエディタで書くことができるが、クリック領域や時間の設定を動画に合わせるのは困難である。そこで、動画を見ながら簡単に動画に合わせた CMML を作成できる CMML オーサリングツールを作成した。CMML の作成を容易にするために、できるだけボタンを押すだけで値を決められるようにし、CMML のプレビューをできるようにした。

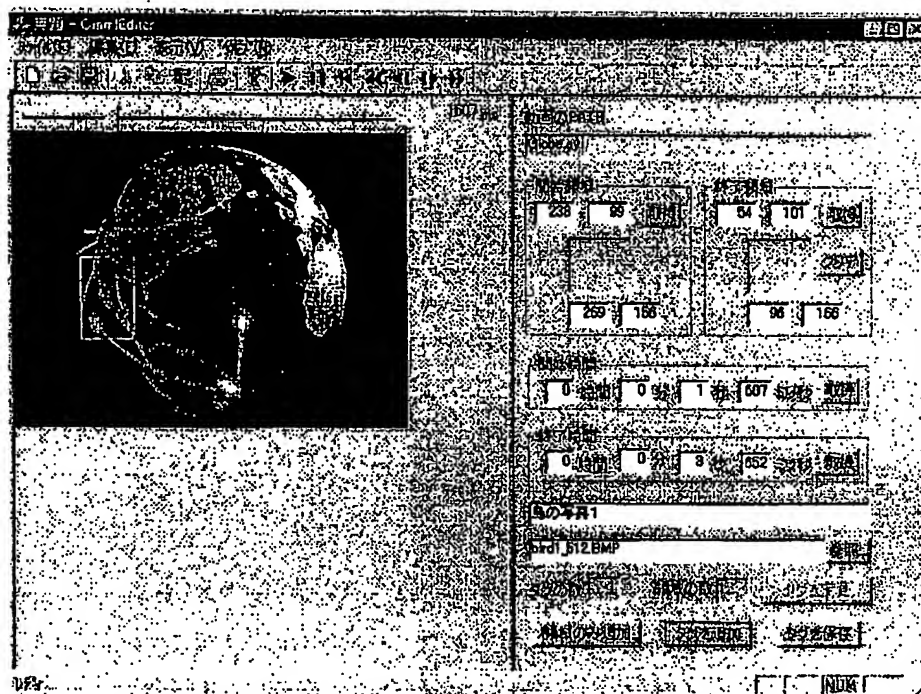


図3 CMML オーサリングツールの実行画面

CMML オーサリングツールは、動画の再生をしながらタグを入力するために、図3のように左右の2つの画面で構成した。画面の左側で動画の再生と CMML のプレビューを行い、画面の右側で CMML タグの値の入力を行う。タグの値の入力は、エディットボックスに直接書くこともできるが、領域と時間においては画面の左側と連動した値を取得することもできるようにした。領域は画面左側でマウスをドラッグした領域を、時間はポジションバーの時間を、それぞれ[取得]ボタンを押して取得する。

設定したクリックタグは、[タグの追加]ボタンを押すことで、CMML に追加される。また、複数の領域を一つのクリックタグとして登録する場合は、領域を設定した後、[領域のみ追加]ボタンを押して領域をクリックタグに追加する。これらの操作を繰り返し行い、全てのクリックタグの追加が終わったら、[タグを保存]ボタンを押して CMML を保存する。

5. おわりに

クリッカブルムービーとして、動画をインタラクティブに扱えるようにした。また、クリッカブルムービーに必要なデータを XML 形式のファイルにすることで、既に保存された動画もクリッカブルムービーとして扱えるようにした。クリッカブルムービーを実現するための XML として CMML を提案し、CMML プレイヤと CMML を作成する為のオーサリングツールを作成した。

本研究の応用として、録画されたテレビ番組や映画に対応する CMML を作成することで、画面上の物を押すと商品情報を見たり、注文ページを表示する等の付加情報サービスが行えと考えられる。また、授業の内容に合わせた CMML を作成し、授業で再生しながら説明することで、教育用の教材としての利用も考えられる。

現在のクリッカブルムービーの問題点として次のようなものが考えられる。CMML プレイヤで表示可能な動画形式が、AVI 形式のみであること。情報ダイアログで表示可能な画像が BMP 形式のみであること。また、クリッカブル領域の移動に関して、直線的な一回の移動しかサポートしていないので、複雑な動きに合わせたクリッカブル領域を作成するのが困難であること等が挙げられる。

これらの問題点に対し、CMML プレイヤで再生できる動画形式を増やし、情報ダイアログで、動画やホームページ等の画像以外の情報を表示できるようにする必要がある。また、複雑な動きのクリッカブル領域をサポートするために、CMML を拡張する必要がある。

さらに、より簡単に複雑なクリッカブル領域のある CMML を作成できるように、CMML オーサリングツールにおいて、マウスでなぞった領域を自動で region タグに変換する機能を追加するのがよいと思われる。また、動画と動画を結びつけるために、CMML の再帰呼び出しを実現すべきだと思われる。ブラウザのプラグインにすることで、インターネットを利用した動画による情報交換ができると思われる。

参考文献

- [1] Simon St. Laurent 著、“よくわかる XML の基礎”、日経 BP 社、1999
- [2] 富士通 XML 推進チーム編、“はじめての XML”、日経 BP 社、1997
- [3] Sean McGrath 著、“XML 開発事例”、アスキー出版局、1999
- [4] Michael Leventhal 他著、“XML 設計技法”、アスキー出版局、1999

付録

CMML の定義

```
<!ELEMENT cmml (moviefile, clickable*)>
<!ELEMENT moviefile (title?)>
<!ATTLIST moviefile href CDATA #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT clickable (start, end, info)>
<!ELEMENT start (region+, time)>
<!ELEMENT region (left, top, right, bottom)>
<!ELEMENT left (#PCDATA)>
<!ELEMENT top (#PCDATA)>
<!ELEMENT right (#PCDATA)>
<!ELEMENT bottom (#PCDATA)>
<!ELEMENT time (hour, min, sec, msec)>
<!ELEMENT hour (#PCDATA)>
<!ELEMENT min (#PCDATA)>
<!ELEMENT sec (#PCDATA)>
<!ELEMENT msec (#PCDATA)>
<!ELEMENT end (region*, time)>
<!ELEMENT info (#PCDATA)>
<!ATTLIST info href CDATA #IMPLIED>
```


**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.